WebObjects Security

Session 715

WebObjects Security

David Neumann
Engineer

Introduction

- Privacy
- App Resource Protection
- Authentication
- Persistent data integrity
- Access control

Keeping It Private

SSL

- SSL is public key crypto on the web
  - Your web server presents its Digital ID
  - Browser checks ID for trusted CA
  - Browser encrypts secret w/ server's public key
  - Only server's private key can decrypt secret
  - All further exchanges encrypted w/ secret key encryption

Using SSL in WebObjects

- If entire site only accessed over SSL, the developer is done

Using SSL in WebObjects

- WO generates partial URLs by default

  /cgi-bin/WebObjects/App.woa/wo/xxxxxxx/1.2

- To force SSL you need to force WO to generate full URLs

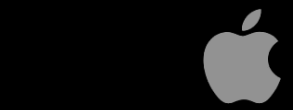
  https://wosite.com/cgi-bin/WebObjects/App.woa/wo/F00000EXSA/1.2

Going From http://to https://

- Create custom WOHyperlink and WOForm components
- Use a redirect technique

SSL URLs Via Custom Component

- Lets you control grain of secure URLs on a page
- Forces request to pass over SSL
- May require custom form elements
- Best for protecting client information



SSL URLs Via Custom Component

```
HyperlinkContainer: WOGenericContainer {
    elementName = "a";
    invokeAction = action;
    href = href;    }
```

- 'action' is the method on your page to invoke
- 'href' is the actual URL WO generates
- Example href:

```
public String href(){
    return "https://hostname" +
        context().componentActionURL();    }
```

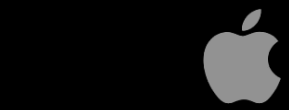
Demo

Using a Custom Component to Generate Absolute URLs

David Neumann
Engineer

SSL Access Via Redirection

- You do not need special components
- Protects the page returned (and subsequent interactions)
- Form values (if any) passed in the clear before redirect
- Best for protecting server information



SSL Access via Redirection

- Override **appendToResponse**
- Let the page determine privacy
- Check if coming in over https://
- If not, redirect back to the page
    - Do not double generate page
    - Do not double invoke the same action

Detecting SSL

- It is tricky; depends on your web server
  - Is there a header named "**HTTPS**" set to "**ON**"?
  - If not, is the server port (header named "**SERVER_PORT**" or "**x-webobjects-server-port**") set to "**443**" ?
  - Other criteria?

Demo

SSL Redirect Without Side Effects

David Neumann
Engineer

Session Hijacking

- What is it?
  - Only care if there is a session AND the user has authenticated to that session
  - SSL alone not enough to protect you
  - The over-the-shoulder attack
- One solution . . .
  - Emit a cookie specific to a user's session
  - If the cookie is missing or does not belong to the session, access to the application is rejected

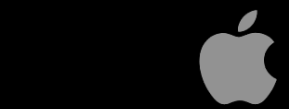
Token Summary

- SessionID
  - Good sessionTimeout sec
- Hijack Cookie (best emitted/sent only over SSL)
  - Good sessionTimeout sec
- User ID/Browser ID Cookies unsecured for personalization
  - Good into distant future
- User ID Cookie only emitted/sent after successful authentication over SSL
  - Good into distant future

Protecting WO Resources

Resource Protection

- What can/cannot be done/seen by the anonymous user
- Types of resource protection
    - Page generation
    - Page creation
    - DirectAction invocation
    - ComponentAction invocation
    - Arbitrary request dispatch



Page Generation Protection

- You let the page get created, but do not let it generate a response
- Override WOComponent's **appendToResponse()**
  - Not necessarily OK to go to a destination
  - Prevent page display no matter how page is accessed
    - Initial app access, DA, or CA

Page Creation Protection

- You override **pageWithName()** on the Application to prevent page creation
  - Might have a list of forbidden page names

Pros/Cons

A. Page creation protection

1. (+) Prevents side-effects in the page constructor from executing unless you are logged-in
2. (–) Cannot affect pages after they are created
3. (–) Your code would have to tolerate getting a different page than expected

B. Page generation protection

1. (+) Allows that page to decide what to do
    1. You might use the same page in different contexts
2. (+) Action code can remain unaltered
3. (–) You have to make sure that constructor code does not require authentication

- I'd argue that B.3 is easier than A.3

DirectAction Protection

- Override **performActionNamed()** to prevent DA invocation
  - DAs can be accessed from anywhere
    - Whether you generate page or not (cannot hide them)
  - Protecting **appendToResponse()** does not prevent the DA from executing
    - . . But does hide the result

Request Level Protection

- You override dispatchRequest() to halt all forms of WO interaction
  - You only have the WORequest to go on
  - You do not know what page will generate if any
  - Could look for required token for any interaction regardless of the request handler used
  - Can use handlerForRequest to figure out what to do per handler type
    - Might require certain form value or cookie to see certain images returned by the WOResourceRequestHandler

ComponentAction Protection

- If you can see a CA, it (usually) means it is OK to execute it
    - If not, do not show it
- In theory you can override WOComponent's invokeAction()
    - But what action would get invoked?
- Backtracking and CAs

Backtracking

- Do not really apply to DAs, only CAs
  - If you should not backtrack to a DA, then it probably should not be a DA
- Can detect with context ID
- Can override invokeAction() on pages where backtracking not allowed
  - Can abort call to super.invokeAction() if backtrack detected
  - Results in merely page refresh

Demo

Page Generation Protection/ DA Invocation Protection

David Neumann
Engineer

Authentication

When to Login

- Front Gate
- On demand

Front Gate Protection

- Usual form
- Somewhat unfriendly
- Users cannot vector in

Browse Then Demand

- Allow surfing until login required
- Two ways to go . . .
    - Go to login page, and re-navigate to protected page
    - Prompt for login then immediately access protected resources
- Combine with the resource protection schemes

Demo

Page & Action Protection
On-Demand Login
Backtrack Detection

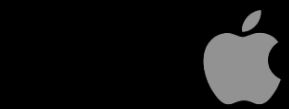
David Neumann
Engineer

Sessionless Login

- Benefits
  - Allows login page to be bookmarked
  - No “session expired” on login!
  - Less resource impact on you
- For HTML page, use WOForm and DirectAction
- Use the DA action handler as the “default action handler”

Authentication Methods

- HTML page
- HTTP panel
- Client Certificates
- Biometric
- Combinations
- Other . . .



Using HTTP Challenge Panel

- You must emit certain status and headers in responses
- You must look for certain headers in requests
- Your web server might not work
- You have to parse Base64-encoded data

Using HTTP Challenge Panel

- Your web server must use an interface that passes the "authorization" header to the WOAdaptor
- What is the realm!?
  - A name associated with resources behind the web server
  - Your application and the web server might use the same realm name
  - Can provide element of SSO for static HTML and WOApps

Using Client-Side Certificates

- Need special www server setup
- Credentials are a token
    - Your private/public key pair
    - Can be stored on hardware device
- Can provide SSO for WOApps and static HTML

Using Client-Side Certificates

- Sort of the inverse of regular login
  - SSL proves you are who you say
  - The application decides if the person is in the system
- Need to modify WOAdaptor source
  - To request certificates from interface if needed
  - To pass the client certificates on to WOApp as header

Verifying Certificates

- See if the certificate has been revoked
  - Download a CRL from CA's web server
  - Get status online using VA's OCSP server

Using Biometrics

- Something about you . . .
  - Physical imaging
    - Thumbprint, palmprint, retina, face
  - Your voice
  - How you type
- Combination with password

Demo

Biometric Login Example

David Neumann
Engineer

Recording Login Failures

- Write out row for failed access attempts
  - Allows for auditing reports
  - Required for multiple instances
- Fetch against table and do count on attempts
  - By username, IP, and/or cookie (in last X seconds)
- Use count to drive counter measures
  - Emailing admin
  - Disabling an account
  - Disabling access from certain IP

Demo

Denying Login due to Intrusion Attempts

David Neumann
Engineer

Persistent Data Integrity

Hashing

For Ex: Passwords

- Do not ever store passwords in the clear
- Storing a password hash is better
- Hash incoming passwords and compare to the stored hash
- If someone cracks your DB, the “passwords” are useless (sort of)
    - BTW: What is a hash?

Encryption

For Ex: Credit Card Numbers

- If item must be recovered (and not just equality tested)
- Use special accessors to do this transparently
- You might let a hardware device perform the actual encryption/decryption (PKCS#11)
    - The key never leaves the device (gotta hack the device not just the computer)
        - Chrysalis-ITS
        - nCipher

Demo

Using Hashes and Encryption

David Neumann
Engineer

Signatures and Timestamps

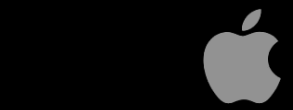
For Ex: Documents

- Store hashes of docs with stored doc
  - Easy to alter doc and rehash
- Store digitally signed docs
  - Cannot alter doc then fake a rehash
- Embed digital timestamp with signed doc
  - Now we have proof when it was created
- Format for the doc is ideally XML
  - Standard DTD exists

http://www.oasis-open.org/cover/dridtd-19990119.html

Nonrepudiation

- You have it if you can prove an event happened
  - In the paper world, it is via ink signatures
  - In the electronic world, it is via digital signatures



What Is a Digital Signature?

- You hash a message
- You use your private key to encrypt the hash
- You append the encrypted hash to the message

What Is a Digital Timestamp?

- A way to prove that not only was something signed, it was signed on a certain date
- Allows a document signature to be considered valid even if Digital ID is revoked later
- Timestamping provided by third parties
    - Surety, ValiCert, e-TimeStamp.com

Messaging Integrity

B2C Digital Signatures

- Clients require a browser plug-in or applet
- Example applications
    - Employee forms processing
    - Brokerage enrollment
    - Paperless workflow with authorization

B2B Digital Signatures

- When machines send and receive digitally signed messages
- Ex: DropShip order, PO, any EDI message

B2B Infrastructure in WO

- WODirectActions
  - Turn WOApps into services
- WOMessage
  - Programmatically send WORequests to remote apps; read WOResponses
- XML management
  - Generate XML sent over the net
  - Interpret XML received

Demo

B2B Example of Signing/Validating XML

David Neumann
Engineer

B2B Scenario

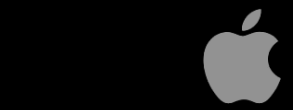
- Acme issues PO to WidgetCo
  - Creates an XML document
  - Signed using the Java's sun.security.* package
  - Encrypted using WidgetCo's public key
  - Sent using WOMessage API

B2B Scenario

- WidgetCo receives PO from Acme
  - Decrypts with private key
  - Verifies Acme digital signature is valid
  - Verifies Acme digital ID is valid
    - Using a CRL or ValiCert VA
  - Creates a “digital receipt” by
    - Combining Acme’s signed request with a “digital timestamp”
    - And signing it all with WidgetCo’s private key
  - Digital receipt returned to Acme

Using Acrobat 5.0 Signatures

- Acrobat 5.0 includes support for digital signatures
    - Viewing them
    - Adding them
    - Verifying them
- Works on Mac OS X, Mac OS 9 and Win32



Demo

B2C Example Using Acrobat 5.0

David Neumann
Engineer

Access Control

Access Control

- Degree of access granted after they login
- Given object A (an EO, say), can user B:
  - See it?
  - Edit it?
  - Do something else with it?
- Access depends on the state of both A and B

Access Control

Techniques

- Have all your EOs implement an interface like this:

```
public boolean canShow(User usr);
public boolean canEdit(User usr);
public Object valueForKeyUser(String key, User usr);
public void takeValueForKeyUser
(Object val, String key, User usr);
```

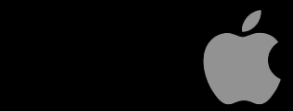
Access Control

Techniques

- Implementation of SecuredEO might be:

```
public boolean canShow(User usr){
    if(usr.equals(creator()))
        return true;
    else if(owners().containsObject(usr))
        return true;
    return false;
}

public boolean canEdit(User usr){
    return canShow(usr);
}
```

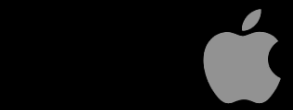


Apply It to the UI

- Use WOConditionals to hide edit controls
- Filter non-readable EOs out of result sets
- Use group membership to hide/show areas of the page
- Create a family of access control aware custom widgets

Attribute Access Control

- Sets of Attributes per Entity
- Different attributes per EO



Complex Access Control

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)

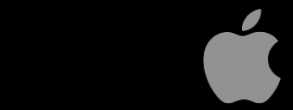
Complex Access Control

Ex: Discretionary Access Control

- Your SecureEOs might have relationships like these:
  - owners():To-many to a set of User objects
  - groups(): To-many to a set of Group objects
  - permission(): To-one to a Permission object

Unix DAC vs. ACL DAC

- Unix file system DAC is a small subset of an ACL DAC
- It is tricky to grant access to a group and then withhold it from a member
- ACLs that result in a Unix DAC
  - *.* [permission]
  - @.* [permission]
  - *.[group] [permission]



Complex Access Control

Ex: Mandatory Access Control

- MAC is about limiting info flow rather than privilege to change info
- Instead of permission you have secrecy Levels
  - "Secret", "Confidential", "Unclassified"
- Instead of groups you have areas of info called Compartments
  - "Accounting", "Shipping", "Marketing"

Complex Access Control

Ex: Mandatory Access Control

- Changes to an EOEditingContext for MAC
  - Implement delegate to intercept object lifecycle calls
  - Will not save unless items have Compartment and Level assigned
  - Will not allow insert if item has a Level < that of user
    - User with Secret permission could not write to a lower permission level like Unclassified

Demo

EO-level and Attribute-level Access Control

David Neumann
Engineer

Summary

- Protecting privacy
  - SSL, components, redirection, detection
- Protecting resources
  - WO request handling overrides
- Authentication
  - Gather credentials via HTML page, certs, panels, etc.

Summary

- Persistent data integrity
  - Using hashing and encryption
- Access control
  - Using ACLs with Eos, Entities

For More Information

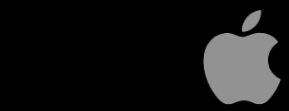
Get the FAQ **http://www.rsa.com/**	HSM Vendor **http://www.ncipher.com/**
Leading VA **http://www.valicert.com/**	Dig Receipt **http://www.oaisi-open.org/**
Leading CA **http://www.verisign.com/**	Dig Timestamp **http://www.verisign.com/**
Signing Plug-in **http://www.adobe.com/**	

For More Information

- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  www.apple.com/services/technicalsupport
- Other places
  www.apple.com/webobjects
  developer.apple.com/webobjects
  www.apple.com/services
  www.info.apple.com/webobjects
- Subscribe to
  webobjects-announce@apple.com

How to Access Documentation

- Most up-to-date: PDF and HTML
  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand
  Vervante.com under Related Resources
- Product CD
  Documents folder and installed in /Developer/Documentation/WebObjects
- In the box (localized)
  Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications, Discovering WebObjects for HTML
- Check ADC News for latest updates
  http://developer.apple.com/devnews



WebObjects Beta

- To be considered for the beta

  Appleseed.apple.com/webobjects

WebObjects Lab

- Located downstairs in Room L
- Lab hours
  - Monday 12:00pm–6:00pm
  - Tuesday 9:00am–2:00pm
  - Wednesday 9:00am–6:00pm
  - Thursday 9:00am–6:00pm
  - Friday 9:00am–6:00pm

Roadmap

FF013 WebObjects

Room A1
Fri., 3:30pm

Who to Contact

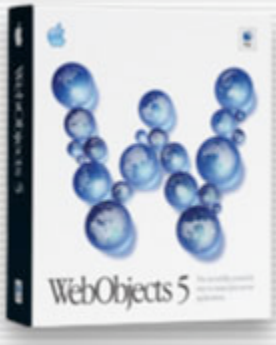
Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

Apple Professional Services (Training, Support, Consulting)
(800) 848-6398
services@apple.com

http://developer.apple.com/wwdc2002/urls.html

Q&A



Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002